DRAFT SF 298

1. Report Date (dd-mm-yy)	2. Report Type	3. Dates covered (from... to)
	1995	

4. Title & subtitle A Fixed-Order, Mixed H2/L1 Control Synthesis Method for Continuous Linear Systems	5a. Contract or Grant #
	5b. Program Element #
6. Author(s)	5c. Project #
	5d. Task #
	5e. Work Unit #

7. Performing Organization Name & Address	8. Performing Organization Report #
9. Sponsoring/Monitoring Agency Name & Address	10. Monitor Acronym
	11. Monitor Report #

12. Distribution/Availability Statement
Distribution approved for Public Release, Distribution Unlimited

13. Supplementary Notes TR downloaded from a WWW unrestricted URL.

14. Abstract

15. Subject Terms

Security Classification of			19. Limitation of Abstract	20. # of Pages	21. Responsible Person (Name and Telephone #)
16. Report Unclass	17. Abstract Unclass	18. This Page Unclass	Unlimited		

A FIXED-ORDER, MIXED H_2/L_1 CONTROL SYNTHESIS METHOD FOR CONTINUOUS LINEAR SYSTEMS*

Mark S. Spillman†and D. Brett Ridgely‡
Air Force Institute of Technology/ENY
Dept. of Aeronautics and Astronautics
2950 P Street
Wright-Patterson AFB, OH 45433-7655

Abstract

This paper discusses a new fixed-order (sub-optimal) H_2/ℓ_1 control synthesis method for continuous linear systems. This new optimization method allows the control system designer to combine H_2 and L_1 norms of dissimilar transfer functions into a single multi-objective optimization problem. Two numerical approaches of approximating the L_1norm and analytical gradients for the H_2 and L_1 problems are presented. The H_2/L_1 optimization method is demonstrated on a realistic aircraft control problem.

1 Introduction

In the past few years, much attention has been paid to the problem of solving multi-objective optimal control problems. The main impetus for this type of research is the simple fact that a real system is never limited to a single class of inputs or outputs. For example, it is not uncommon in many aircraft problems to have some inputs well modeled as white Gaussian noise, some as bounded energy and some as bounded magnitude signals. While different control methodologies exist to handle each of these classes of inputs separately, multi-objective optimization methods allow the designer to optimally handle the trade-offs associated with several classes of inputs at one time.

Much of the early work in multi-objective optimal control has focused on H_2/H_∞ optimization methods. Many of these approaches are restricted to special cases such as one input/one output, one input/two outputs and two inputs/one output.[1–7] The general H_2/H_∞ two exogenous inputs two controlled outputs problem was first solved by Khargonekar and Rotea[7] using full state feedback. Since the optimal solution to the problem is in general non-rational,[8,9] several researchers have recently developed sub-optimal solution methods. Walker and Ridgely[10] developed a numerical algorithm to solve the general fixed order H_2/H_∞ problem with output feedback. Their approach also allows for both singular and multiple H_∞ constraints.[11] Ly and Schömig[12] solve the identical problem using gradient based parameter optimization methods.

The L_1 optimization problem was first proposed by Vidyasagar[13] to optimally reject persistent bounded magnitude inputs. Dahleh and Pearson[14] later developed exact and approximate methods of solving the general single input, single output L_1 problem. Unfortunately, both approaches lead to irrational controllers. A few researchers have incorporated a discrete version of the L_1 problem (denoted as ℓ_1 optimization) into discrete multi-objective optimization methods,[15–17] but until now, no one has incorporated the L_1 problem into a continuous time, multi-objective problem.

The purpose of this paper is to present a numerical approach to the continuous time, fixed-order H_2/L_1 optimization problem. The remainder of this paper is organized as follows. Section 2 describes the complete set-up of the fixed-order H_2/L_1 optimization problem. In Section 3, analytic gradients for the H_2 problem are presented. Two numerical methods of finding approximate solutions to the L_1 problem are presented in Section 4 and analytic gradients for the L_1 problem are discussed in Section 5. Section 6 covers computer implementation issues and the method is applied to an AFTI F-16 longitudinal control problem in Section 7. Finally, Section 8 sum-

†Email: mspillma@falcon.flight.wpafb.af.mil.
‡Email: dridgely@afit.af.mil.

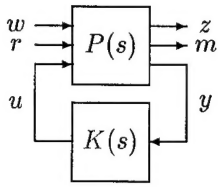

Figure 1: Mixed H_2/L_1 optimization problem

marizes the paper and presents some conclusions.

2 The Mixed H_2/L_1 Problem

The mixed H_2/L_1 design problem is depicted in Figure 1. In general, it is assumed that there is no relationship between w and r, or z and m.

The goal of mixed H_2/L_1 optimization is to find a stabilizing controller which satisfies

$$\inf_{K\text{stabilizing}} \|T_{zw}\|_2 \quad \text{subject to} \quad T_{mr}\|_1 \leq \nu \qquad (2.1)$$

where

$$T_{zw} = \left[\begin{array}{c|c} \mathcal{A}_2 & \mathcal{B}_w \\ \hline \mathcal{C}_z & \mathcal{D}_{zw} \end{array}\right], \qquad (2.2)$$

$$T_{mr} = \left[\begin{array}{c|c} \mathcal{A}_1 & \mathcal{B}_r \\ \hline \mathcal{C}_m & \mathcal{D}_{mr} \end{array}\right] \qquad (2.3)$$

and ν is a user specified constraint level on the L_1 norm of T_{mr}.

The state-space of P is found by augmenting the stable weights of the H_2 problem and the L_1 problem to the original system. Typically, the orders of the individual H_2 and L_1 problems are less than the order of P. The state-space equations of the H_2 and L_1 problems can be written as

$$\begin{aligned} \dot{x}_2 &= A_2 x_2 + B_w w + B_{u2} u \\ z &= C_z x_2 + D_{zw} w + D_{zu} u \\ y &= C_{y2} x_2 + D_{yw} w + D_{yu} u \end{aligned} \qquad (2.4)$$

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_r r + B_{u1} u \\ m &= C_m x_1 + D_{mr} r + D_{mu} u \\ y &= C_{y1} x_1 + D_{yr} r + D_{yu} u \end{aligned} \qquad (2.5)$$

where x_2 is the state vector for the underlying H_2 problem, and x_1 is the state vector of the underlying L_1 problem. The following assumptions are now made on the state-space elements in Equation 2.4 and Equation 2.5.

1. (A_2, B_{u2}) stabilizable, (C_{y2}, A_2) detectable
2. (A_1, B_{u1}) stabilizable, (C_{y1}, A_1) detectable
3. $D_{zw} = 0$
4. $D_{zu}^T D_{zu}$ full rank, $D_{yw} D_{yw}^T$ full rank
5. $\left[\begin{array}{cc} A_2 - j\omega I & B_{u2} \\ C_z & D_{zu} \end{array}\right]$ has full column rank for all ω
6. $\left[\begin{array}{cc} A_2 - j\omega I & B_w \\ C_{y2} & D_{yw} \end{array}\right]$ has full row rank for all ω
7. $D_{yu} = 0$

Assumptions 1-2 ensure that stabilizing controllers for the H_2 and L_1 problems can be found. Assumption 3 is necessary for T_{zw} to have a finite two-norm and Assumption 4 keeps the problem from being singular. Assumptions 5-6 are required for the existence of stabilizing solutions to the algebraic Ricatti equations used in the H_2 solution. Finally, Assumption 7 is not required but it simplifies the notation in the following development.

The controller state-space equations are

$$\begin{aligned} \dot{x}_k &= A_k x_k + B_k y \\ u &= C_k x_k + D_k y \end{aligned} \qquad (2.6)$$

Using Equation 2.4 and Equation 2.6, the closed-loop state-space equations for T_{zw} can be written as

$$\begin{aligned} \dot{x}_2 &= (A_2 + B_{u2} D_k C_{y2}) x_2 + (B_{u_2} C_k) x_k \\ &\quad + (B_w + B_{u2} D_k D_{yw}) w \\ \dot{x}_k &= (B_k C_{y2}) x_2 + (A_k) x_k + (B_k D_{yw}) w \\ z &= (C_z + D_{zu} D_k C_{y2}) x_2 + (D_{zu} C_k) x_k \\ &\quad + (D_{zu} D_k D_{yw}) w \end{aligned} \qquad (2.7)$$

Notice that $D_{zu} D_k D_{yw}$ in Equation 2.7 must be zero to ensure the resulting two-norm of T_{zw} is finite. This fact and Assumption 4 imply that D_k must be zero. Therefore, a strictly proper controller, K, can be assumed without any loss of generality. With this

additional assumption, the closed state-space matrices in Equations 2.2-2.3 can be written as

$$\mathcal{A}_2 = \begin{bmatrix} A_2 & B_{u2}C_k \\ B_kC_{y2} & A_k \end{bmatrix}$$
$$\mathcal{A}_1 = \begin{bmatrix} A_1 & B_{u1}C_k \\ B_kC_{y1} & A_k \end{bmatrix}$$

$$\mathcal{B}_w = \begin{bmatrix} B_w \\ B_kD_{yw} \end{bmatrix} \quad \mathcal{B}_r = \begin{bmatrix} B_r \\ B_kD_{yr} \end{bmatrix}$$

$$\mathcal{C}_z = \begin{bmatrix} C_z & D_{zu}C_k \end{bmatrix} \quad \mathcal{C}_m = \begin{bmatrix} C_m & D_{mu}C_k \end{bmatrix}$$

$$\mathcal{D}_{zw} = 0 \quad \mathcal{D}_{mr} = D_{mr}$$

The following definitions are used to discuss the solution to the mixed H_2/L_1 problem:

$$\begin{aligned}
\underline{\alpha} &:= \inf_{K\text{admissable}} \|T_{zw}\|_2 \\
\underline{\nu} &:= \inf_{K\text{admissable}} \|T_{mr}\|_1 \\
K_{2_{opt}} &:= \text{the unique } K \text{ that makes } \|T_{zw}\|_2 = \underline{\alpha} \\
K_{1_{opt}} &:= \text{a } K \text{ that makes } \|T_{mr}\|_1 = \underline{\nu} \\
\overline{\alpha} &:= \|T_{zw}\|_2 \text{ when } K = K_{1_{opt}} \\
\overline{\nu} &:= \|T_{mr}\|_1 \text{ when } K = K_{2_{opt}} \\
K_{mix} &:= \text{the global minimum solution to the} \\
&\quad H_2/L_1 \text{ problem for some } \nu \\
\alpha &:= \|T_{zw}\|_2 \text{ when } K = K_{mix} \\
\nu &:= \|T_{mr}\|_1 \text{ when } K = K_{mix}
\end{aligned}$$

Admissible controllers must be stabilizing and have a fixed, user-specified order.

A solution to the H_2/L_1 problem must satisfy the Kuhn-Tucker necessary conditions:

1. K_{mix} must be feasible, i.e. it must stabilize T_{zw} and T_{mr}
2. $\nabla\|T_{zw}\|_2 + \lambda_1\nabla\|T_{mr}\|_1 = 0, \quad \lambda_1 \geq 0$
3. $\lambda_1(\|T_{mr}\|_1 - \nu) = 0, \quad \lambda_1 \geq 0$

where λ_1 is a Lagrange multiplier associated with the one-norm constraint. Condition 1 is simply a feasibility condition. Condition 2 states that the gradient of the objective must be balanced by the scaled gradient of the constraint. The last condition states that if the constraint is not satisfied exactly, then λ_1 must be zero. Conditions 2 and 3 together imply that an optimal solution (if it exists) must lie on the constraint boundary when $\underline{\nu} \leq \nu \leq \overline{\nu}$. If $\nu \geq \overline{\nu}$, then the unique solution is $K_{2_{opt}}$. By definition, ν can not be chosen less than $\underline{\nu}$. When $\underline{\nu} \leq \nu \leq \overline{\nu}$, α is a monotonically decreasing function of ν.

A sequential quadratic programming (SQP) algorithm is used to solve the H_2/L_1 problem numerically. The objective, f, and the constraints, g, are given by

$$\begin{aligned}
f(c) &= \zeta_2\|T_{zw}\|_2^2 \\
g_1(c) &\leq \zeta_1(\|T_{mr}\| - \nu) \\
g_s(c) &\leq \zeta_s\left[\ \max_i(Re(\lambda_i(\mathcal{A}_2)))\ \right] \qquad (2.8)
\end{aligned}$$

where c is a vectorized compensator, ζ's are scaling parameters, and $\lambda_i(\cdot)$ is the i^{th} eigenvalue of $(\cdot)$. The stability constraint, g_s, is added to the problem to keep the SQP algorithm from getting lost in the unstable region. While this constraint should be posed as a strict inequality constraint, equality must also be allowed to incorporate it into the SQP algorithm. Modal form is assumed for the controller, K, to minimize the number of design variables in c. While this approach disallows repeated eigenvalues in the controller, it has been shown to be sufficient in practice.

The SQP algorithm requires gradients for the objective function and each of the constraints. Analytical gradients for the objective function are derived in the next section.

3 Gradients of the Two-Norm

The two-norm squared of T_{zw} can be calculated by

$$\|T_{zw}\|_2^2 = tr[QC_z^TC_z] \qquad (3.1)$$

where Q is the positive semidefinite solution to the Lyapunov equation

$$\mathcal{A}_2Q + Q\mathcal{A}_2^T + \mathcal{B}_w\mathcal{B}_w^T = 0 \qquad (3.2)$$

If the constraint on Q in Equation 3.2 is written in an equivalent form

$$tr[(\mathcal{A}_2Q + Q\mathcal{A}_2^T + \mathcal{B}_w\mathcal{B}_w^T)X] = 0 \ \ for\ \ all\ \ X \qquad (3.3)$$

then

$$\begin{aligned}
\|T_{zw}\|_2^2 &= tr[QC_z^TC_z] + \ tr[(\mathcal{A}_2Q + Q\mathcal{A}_2^T + \mathcal{B}_w\mathcal{B}_w^T)X] \\
&\quad for\ all\ X. \qquad (3.4)
\end{aligned}$$

To simplify the notation, let J equal the two-norm squared of T_{zw}. Notice from Equation 3.4 that J is an explicit function of the design variables and the

matrices, X and Q. However, the partial derivative of J with respect to X is simply the left side of Equation 3.2 and, therefore, it is always equal to zero. Further, it can be shown that

$$\frac{\partial J}{\partial Q} = \mathcal{A}_2^T X + X\mathcal{A}_2 + \mathcal{C}_z^T \mathcal{C}_z = 0 \qquad (3.5)$$

which is a Lyapunov equation. Since X is arbitrary, let X be the positive semidefinite solution to the Lyapunov equation

$$\mathcal{A}_2^T X + X\mathcal{A}_2 + \mathcal{C}_z^T \mathcal{C}_z = 0 \qquad (3.6)$$

This choice of X is guaranteed to exist since A_2 must be stable. With this choice, $\frac{\partial J}{\partial Q} = 0$ and the only remaining derivatives to be calculated are $\frac{\partial J}{\partial A_k}$, $\frac{\partial J}{\partial B_k}$, and $\frac{\partial J}{\partial C_k}$.

If Q is given by

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \qquad (3.7)$$

and X is given by

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \qquad (3.8)$$

then these derivatives are

$$\begin{aligned} \frac{\partial J}{\partial A_k} &= 2X_{12}^T Q_{12} + 2X_{22}Q_{22} \\ \frac{\partial J}{\partial B_k} &= 2X_{12}^T Q_{11} C_{y2}^T + 2X_{22}Q_{12}^T C_{y2}^T + \\ &\quad 2X_{12}^T B_w D_{yw}^T + 2X_2 B_k D_{yw} D_{yw}^T \\ \frac{\partial J}{\partial C_k} &= 2B_{u2}^T X_{11} Q_{12} + 2B_{u2}^T X_{12} Q_{22} + \\ &\quad 2D_{zu}^T C_z Q_{12} + D_{zu}^T D_{zu} C_k Q_{22} \end{aligned} \qquad (3.9)$$

The complete method for finding the partial derivatives of J with respect to the vectorized compensator, c, is as follows

1. Solve the Lyapunov equations in Equation 3.2 and Equation 3.6 for Q and X respectively.
2. Compute the partial derivatives in Equation 3.9.
3. Rearrage Equation 3.9 to express the gradient with respect to the vector of design variables, c, as a vector

$$\frac{\partial J}{\partial c} = \begin{bmatrix} \begin{bmatrix} \left(\frac{\partial J}{\partial A_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial A_k}\right)_{n_k}^T \end{bmatrix}^T \\ \begin{bmatrix} \left(\frac{\partial J}{\partial B_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial B_k}\right)_{n_y}^T \end{bmatrix}^T \\ \begin{bmatrix} \left(\frac{\partial J}{\partial C_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial C_k}\right)_{n_k}^T \end{bmatrix}^T \end{bmatrix} \qquad (3.10)$$

where the individual vectors in parentheses are the columns of the partial derivative matrices, n_k is the order of the compensator, and n_y is the number of measurements. The next section discusses two numerical methods of approximating the L_1 norm of T_{mr}.

4 Calculating the L_1 Norm

The L_1 norm of a continuous, SISO system is given by

$$\|T_{mr}\|_1 = \int_0^\infty \mid \mathcal{C}_m e^{\mathcal{A}_1 t}\mathcal{B}_r \mid dt + \mid \mathcal{D}_{mr} \mid \qquad (4.1)$$

If $\mathcal{A}_1$ is stable, then $e^{\mathcal{A}_1 t}$ approaches zero as t approaches infinity, and Equation 4.1 can be approximated by truncating the integral at some point in time, t_N, which can be computed from the eigenvalues of $\mathcal{A}_1$. The remaining question is how best to evaluate the truncated integral.

The first approach to this problem is to eliminate the absolute value sign inside the integral by determining where the impulse response is positive and negative. This can be done by discretizing the continuous system with a zero order hold (ZOH), and finding the pulse response of the resulting discrete system. Since the ZOH transformation preserves the values of the continuous impulse response at the sample points, the discrete pulse response can be used to find approximate locations where the continuous impulse response is zero. These approximate locations are then refined to any degree of accuracy by using a nonlinear root-solver on the continuous function, $\mathcal{C}_m e^{\mathcal{A}_1 t}\mathcal{B}_r$. Once the zero locations of the impulse response are known, the absolute value sign can be removed by breaking the integral in Equation 4.1 into a series of integrations. Further, if $\mathcal{A}_1$ is invertible (i.e. $\mathcal{A}_1$ has no zero eigenvalues), then

$$\int_{t_1}^{t_2} \mathcal{C}_m e^{\mathcal{A}_1 t}\mathcal{B}_r dt = \mathcal{C}_m \mathcal{A}_1^{-1}[e^{\mathcal{A}_1 t_2} - e^{\mathcal{A}_1 t_1} - 2I]\mathcal{B}_r \qquad (4.2)$$

Thus, each integral in the series can be determined exactly. When $\mathcal{A}_1$ does have eigenvalues at the origin, the system is not stable and the one-norm is infinite.

The key to this method rests on determining the zero locations of the continuous impulse response. If zero locations are missed in the discretization step due to high frequency dynamics, then the one-norm

will be inaccurate. In addition, most nonlinear root-solvers are only capable of finding the root closest to a given initial guess. This implies that the approximate root locations must be fairly precise, i.e. the discretization sample period must be fairly small. Using a small sample period to estimate the impulse response of systems with fast and slow dynamics can be expensive in terms of computer time. These issues make the above method impractical to use in a computer algorithm which must handle a wide range of problems. However, this method does offer an alternative way to calculate the one-norm for specific examples, and can be used to check another method.

The second approach to approximating the truncated integral is more robust but less accurate. In this method, the continuous system is discretized with a ZOH transformation using a small sample period. The truncated integral is then approximated with a trapezoidal integration of the discrete pulse response. As the sample period decreases, the approximation improves. If H_2/L_1 optimization is performed near L_1 optimal, where t_N is small, this method works well at approximating the one-norm of a system without requiring an unreasonable number of samples. Starting points relatively near L_1 optimal can be found by using any fixed-order H_∞ optimization method which allows the structure of the controller to be set by the user.[10,12]

The L_1 norm for MIMO systems is calculated from the maximum row sum of SISO transfer function norms. However, discontinuous gradients can occur when the maximum row sum occurs over more than one row. To counter this problem, each row sum is constrained as a separate Multiple Input Single Output (MISO) transfer function. This effectively adds more constraints to the H_2/L_1 optimization problem, but most of the additional constraints are not active at any specific design point. If an optimization algorithm is used which only calls for gradients of the active constraints, then these additional constraints have little impact on the overall performance of the H_2/L_1 algorithm. The next section discusses how to calculate the gradients associated with the L_1 norm.

5 Gradients of the One-Norm

This section derives the gradients of the one-norm with respect to the design variables in the matrices A_k, B_k, and C_k. Since all of these matrices appear in $\mathcal{A}_1$, the first problem considered is how to compute the partial of $e^{\mathcal{A}_1 t}$ with respect to any element of $\mathcal{A}_1$. This is the most difficult issue in calculating the gradients of the one-norm with respect to the design variables. Contributions to the one-norm gradients from the other closed-loop state-space matrices, which are considerably simpler to find, are discussed thereafter. In solving the first problem, it will also become clear how to calculate the gradient of the added stability constraint in Equation 2.8.

If $\mathcal{A}_1$ is a non-defective matrix ($\mathcal{A}_1$ can be diagonalized), then the partial of $e^{\mathcal{A}_1 t}$ with respect to any element of $\mathcal{A}_1$ is given by

$$\begin{aligned} \frac{\partial e^{\mathcal{A}_1 t}}{\partial a_{ij}} &= \frac{\partial}{\partial a_{ij}} \left[\; R e^{\Lambda t} R^{-1} \; \right] \qquad (5.1) \\ &= \frac{\partial R}{\partial a_{ij}} e^{\Lambda t} R^{-1} + R \frac{\partial e^{\Lambda t}}{\partial a_{ij}} R^{-1} + R e^{\Lambda t} \frac{\partial R^{-1}}{\partial a_{ij}} \end{aligned}$$

where R is the right eigenvector matrix of $\mathcal{A}_1$ and Λ is a diagonal matrix consisting of the eigenvalues of $\mathcal{A}_1$. The partial of $e^{\Lambda t}$ with respect to a_{ij} can be computed easily from

$$\frac{\partial e^{\Lambda t}}{\partial a_{ij}} = \sum_{i=1}^{n_1} \frac{\partial e^{\Lambda t}}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial a_{ij}} \qquad (5.2)$$

provided that the partial derivative of each eigenvalue is known with respect to a_{ij}. Likewise, the partial derivative of R^{-1} with respect to a_{ij} can be easily computed from

$$\frac{\partial R^{-1}}{\partial a_{ij}} = -R^{-1} \frac{\partial R}{\partial a_{ij}} R^{-1} \qquad (5.3)$$

provided that the partial derivative of the right eigenvector matrix with respect to a_{ij} is known.

The partials of an eigenvalue and eigenvector with respect to a_{ij} can be computed from the standard eigenvalue problem if the n_1 eigenvalues of $\mathcal{A}_1$ are distinct. The derivations of these two derivatives involve both the left and right eigenvectors of $\mathcal{A}_1$, and are described in detail by Nelson.[18] Letting $(\cdot)' = \frac{\partial(\cdot)}{\partial a_{ij}}$, the solutions are given by

$$\lambda_i' = L_i^T \mathcal{A}_1' R_i \qquad (5.4)$$

$$R_i' = \sum_{k=1}^{n_1} \beta_k R_k + \beta_i R_1 = V_i + \beta_i R_i \qquad (5.5)$$

where

$$\beta_k = \frac{L_k^T \left[\; R_i \lambda_i' - \mathcal{A}_1' R_i \; \right]}{\lambda_k - \lambda_i}, \quad i \neq k \qquad (5.6)$$

$$\beta_i = Re \left(\; R_i^H M V_i \; \right) - \frac{1}{2} R_i^H M' R_i \qquad (5.7)$$

$$R_i^H M R_i = 1 \tag{5.8}$$

The notation, $(\cdot)^H$, denotes the complex conjugate transpose of $(\cdot)$, and L is the left eigenvector matrix. R_i and L_i refer to the i^{th} eigenvector of R and L, respectively. Note that $\mathcal{A}_1'$ is simply a $n_1 \times n_1$ matrix of zeros with a one in the (i,j) element. The real part of Equation 5.4 provides a method of computing the gradients of the stability constraint in Equation 2.8 if λ_i is the maximum eigenvalue of $\mathcal{A}_1$.

The partial of $e^{\mathcal{A}_1 t}$ with respect to a_{ij} can be completely determined from Equations 5.2-5.8. Using this information, the partial derivative of the one-norm with respect to $\mathcal{A}_1$ can be found element-by-element from

$$\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{A}_{1_{i,j}}} = \int_0^\infty sgn\Big(\mathcal{C}_m e^{\mathcal{A}_1 t} \mathcal{B}_r \Big) \mathcal{C}_m \frac{\partial e^{\mathcal{A}_1 t}}{\partial a_{ij}} \mathcal{B}_r dt \tag{5.9}$$

where $sgn(\cdot)$ is 1, -1, or 0 depending on the sign of $(\cdot)$. The partial derivatives with respect to the other closed-loop matrices are given by

$$\begin{aligned}
\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{B}_r} &= \int_0^\infty sgn\Big(\mathcal{C}_m e^{\mathcal{A}_1 t} \mathcal{B}_r \Big) \Big[\Big(e^{\mathcal{A}_1 t} \Big)^T \mathcal{C}_m^T \Big] dt \\
\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{C}_m} &= \int_0^\infty sgn\Big(\mathcal{C}_m e^{\mathcal{A}_1 t} \mathcal{B}_r \Big) \Big[\Big(\mathcal{B}_r^T e^{\mathcal{A}_1 t} \Big)^T \Big] dt \\
\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{D}_{mr}} &= sgn\Big(\mathcal{D}_{mr} \Big)
\end{aligned} \tag{5.10}$$

From these expressions, the partials of the one-norm with respect to A_k, B_k, and C_k can be expressed as

$$\frac{\partial \|T_{mr}\|_1}{\partial A_{k_{i,j}}} = \Big[\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{A}_1} \Big]_{n_1+i, n_1+j} \tag{5.11}$$

$$\begin{aligned}
\frac{\partial \|T_{mr}\|_1}{\partial B_{k_{i,j}}} &= \sum_{p=1}^{n_k} \sum_{q=1}^{n_1} C_{y1_{j,q}} \Big[\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{A}_1} \Big]_{n_1+i, n_1+j} + \\
&\quad \sum_{p=1}^{n_k} \sum_{q=1}^{n_r} \delta_{p,i} D_{yr_{j,q}} \Big[\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{B}_r} \Big]_{n_1+p,q} \\
\frac{\partial \|T_{mr}\|_1}{\partial C_{k_{i,j}}} &= \sum_{p=1}^{n_1} \sum_{q=1}^{n_k} \delta_{j,q} B_{u1_{p,i}} \Big[\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{A}_1} \Big]_{p, n_1+q} + \\
&\quad \sum_{p=1}^{n_m} \sum_{q=1}^{n_k} \delta_{j,q} D_{mu_{p,i}} \Big[\frac{\partial \|T_{mr}\|_1}{\partial \mathcal{C}_m} \Big]_{p, n_1+q}
\end{aligned}$$

The partial of the one-norm of T_{mr} with respect to c can be formed from the columns of matrices in Equation 5.11 as described in Section 3.

The integrals in Equations 5.9-5.10 are approximated in the same manner as the integral in the one-norm calculation. The upper limit of integration is changed to t_N, and trapezoidal integration is performed over discrete points from the continuous function. The same discretization period is used for the norm calculation and the gradients, which allows the sign factor required in Equations 5.9-5.10 to be computed from the data gathered in evaluating the one-norm.

As mentioned in the previous section, separate MISO gradients are calculated for each row of a MIMO system. This ensures that continuous gradient information is available regardless of where the maximum row sum occurs.

Recall that development of R' and λ' require that the eigenvalues of $\mathcal{A}_1$ be distinct. Clearly this condition can be violated while performing H_2/L_1 optimization. The current algorithm switches to finite difference calculations for gradient information if this occurs. Research is still being conducted to develop ways to handle an $\mathcal{A}_1$ matrix with repeated eigenvalues.

6 Computer Implementation

The H_2/L_1 optimization problem was implemented using the MATLAB^{TM} SQP routine, *constr.m*. Separate subroutines for each norm, constraint and gradient calculation were called from this algorithm. Unfortunately, this routine requires all gradient information regardless of whether or not a constraint is active. FORTRAN^{TM} shells can be written to allow the MATLAB^{TM} subroutines to interface with IMSL^{TM} optimization routines, which eliminate this problem.

The author's algorithms were also recently integrated into a complete MATLAB^{TM} Toolbox for fixed-order, mixed-norm control synthesis.[19] This toolbox (available by anonymous ftp) allows the designer to find a compensator which minimizes the H_2 norm of a transfer function while constraining any combination of H_∞ and/or L_1 (ℓ_1) norms of possibly dissimilar transfer functions to be below specified constraint levels. The toolbox also features a new MATLAB^{TM} SQP routine based on the FORTRAN^{TM} routine NLPQL written by Schittkowski.[20]

7 H_2/L_1 Design Example

A longitudinal controller design for the AFTI F-16 in Figure 2 is used to demonstrate the H_2/L_1 optimization method. The aircraft plant consists of a 4

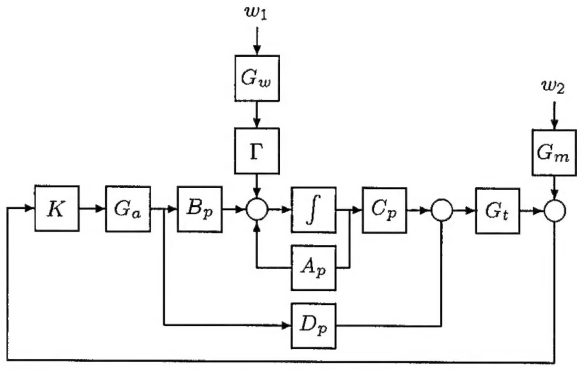

Figure 2: F-16 simulation diagram

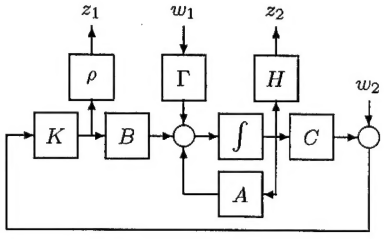

Figure 3: H_2 diagram

state linear model of the aircraft's longitudinal dynamics (G_p), a first order actuator (G_a) and a first order Padé approximation of a 0.05 second time delay (G_t). A_p, B_p, C_p and D_p represent the state-space elements of G_p. For simulation purposes, the disturbances, w_1 and w_2, are added to simulate wind gusts and measurement noise, respectively. G_w is a Von Karmen wind model and Γ is used to allow wind gusts to enter the aircraft plant as an angle-of attack perturbation. G_m is a high-pass filter used to better model sensor noise. Detailed descriptions of all the elements in Figure 2 can be found in the Appendix.

The H_2 problem, taken from an example by Luke,[21] is to find an internally stabilizing controller which minimizes the response of the normal acceleration and weighted control to the wind disturbances and measurement noise. A block diagram of the H_2 problem is shown in Figure 3. A, B and C refer to the state-space matrices of $G = G_a G_p G_t$. The control weight, ρ, equals 10 and the state weighting matrix, H, equals the system C matrix. The wind disturbance, w_1, is modeled as a white Gaussian noise (WGN) with $5.0e-4$ rad^2-sec intensity, and Γ is the column of A corresponding to the angle-of-attack state, α. The measurement noise, w_2, is modeled as a WGN with $1.6e-5$ g^2-sec intensity. Weighted control power, z_1, and the delayed normal acceleration, z_2, are the controlled outputs.

The L_1 problem, depicted in Figure 4, is a weighted sensitivity minimization design. $W_s = \frac{s+10}{s+0.0001}$ is the inverse of the desired sensitivity. A plot of the

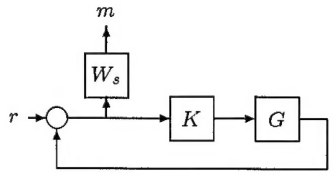

Figure 4: L_1 diagram

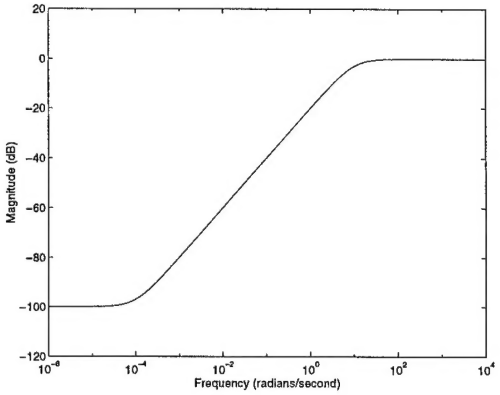

Figure 5: Desired sensitvity

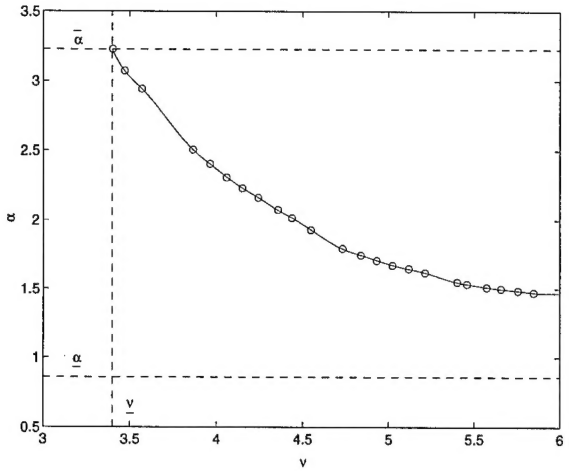

Figure 6: H_2/L_1 solution curve

Table 1: Soultion points

point #	α	ν
1	3.23	3.40
5	2.40	3.96
9	2.07	4.36
15	1.67	5.02
21	1.50	5.66
24	0.86	97,190

desired sensitivity is shown in Figure 5.

7.1 H_2/L_1 Results

A plot of the H_2/L_1 solution curve for a 6^{th} order compensator is shown in Figure 6. Notice that the plot only depicts mixed solutions near L_1 optimal. A complete solution curve would extend much further along the horizontal axis, and asymptotically approach the dashed line, labeled $\underline{\alpha}$. Values for several of the solution points, depicted as circles in Figure 6, are given in Table 1. The points are numbered from left to right in Figure 6. Point number 24 represents the H_2 optimal solution, which is not depicted in Figure 6. The points in Table 1 are referred to as the design points for the remainder of this section.

Plots of the sensitivity and complementary sensitivity for the design points are shown in Figures 7 and 8. In each plot, one curve is dramatically different than the others. This curve corresponds to design point 24, the H_2 solution.

The vector gain margins (VGM) in dB and vector

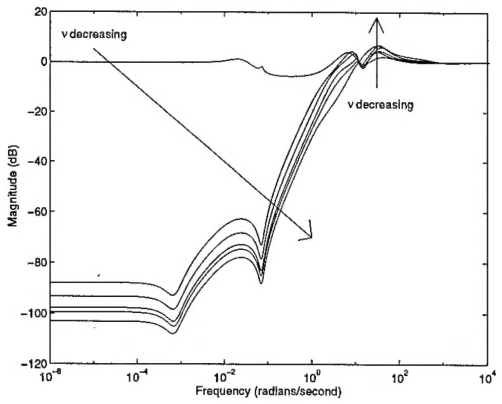

Figure 7: Sensitivity

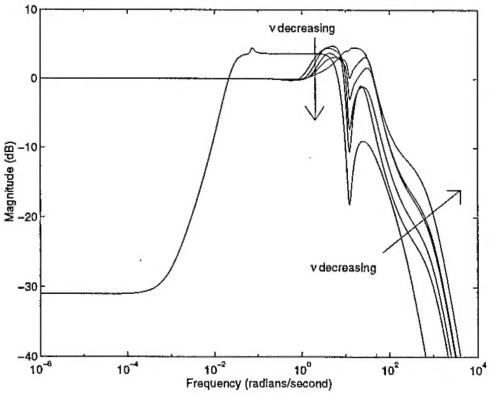

Figure 8: Complementary sensitivity

Table 2: Soultion points

point #	$\underline{VGM}$	$\overline{VGM}$	VPM
1	-7.77	5.42	34.4
5	-10.3	5.42	40.7
9	-9.12	6.10	37.9
15	-7.88	8.02	35.1
21	-7.50	8.11	35.3
24	-7.82	8.80	37.1

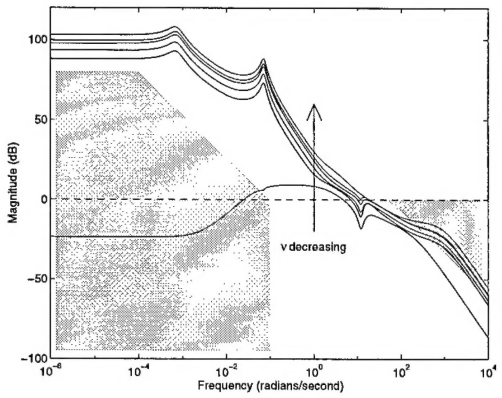

Figure 9: Open-loop GK

phase margins[22] (VPM) in *degrees* for the design points are shown in Table 2. Notice that the stability margins do not consistently improve as ν is decreased. However, the margins are acceptable at all of the design points.

The open-loop GK plot is shown in Figure 9. The shaded area on the left side of Figure 9 represents a recommended performance and disturbance rejection barrier. The shaded area on the right side of Figure 9 represents a recommended sensor noise and unmodeled dynamics barrier. Descriptions of both barriers were taken from Ridgely and Banda,[23] which also contains an excellent discussion of desired GK shapes. Notice that the mixed design points, all relatively near L_1 optimal, miss the barrier on the left, but pass within the barrier on the right. If deemed necessary, a design which meets both barriers could be found by a mixed solution closer to H_2 optimal.

The loop shapes from the design points imply that high frequency noise will be more prevalent than low frequency noise in system responses. Systems obtained from these design points should also track low frequency commands well. Figure 10 shows the

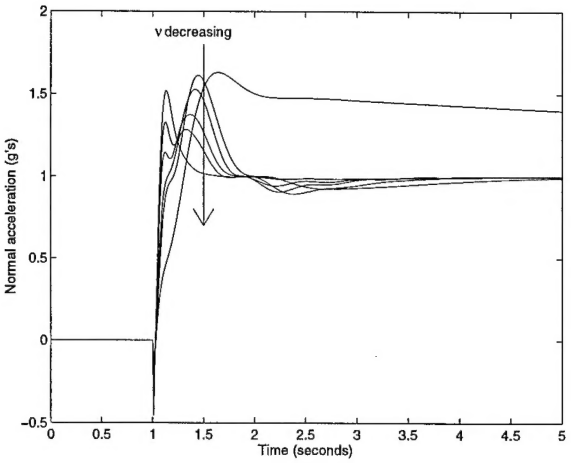

Figure 10: Step responses without noise

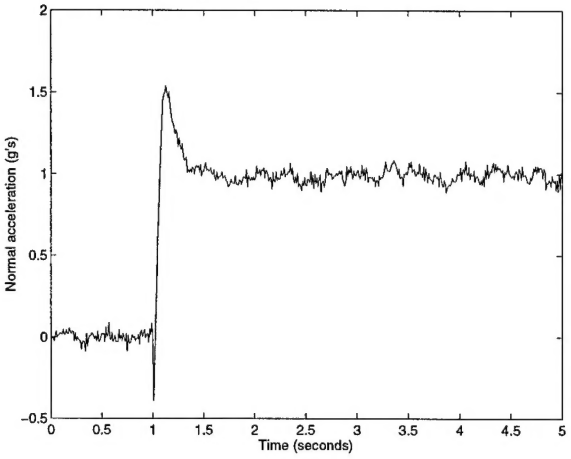

Figure 11: Step response with noise, design point 1

F-16 responses, without noise, to a commanded $1g$ (from trim) step input for the different designs. Notice that the H_2 solution tracks with a steady-state error while the mixed designs do not. Step responses with noise for design points 1, 9, 15 and 24 are shown in Figures 11-14, respectively. As expected, the systems with lower values contain more high frequency noise than those closer to the H_2 optimal design. The most important item to note from Figures 11-14 is that the tracking performance of the AFTI F-16 can be greatly improved using mixed H_2/L_1 designs, with very little increase on the amount of noise in the system response.

8 Conclusions

This paper presented a newly developed numerical

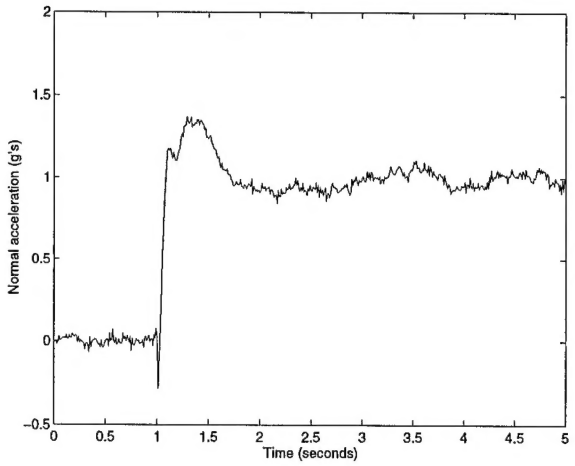

Figure 12: Step response with noise, design point 9

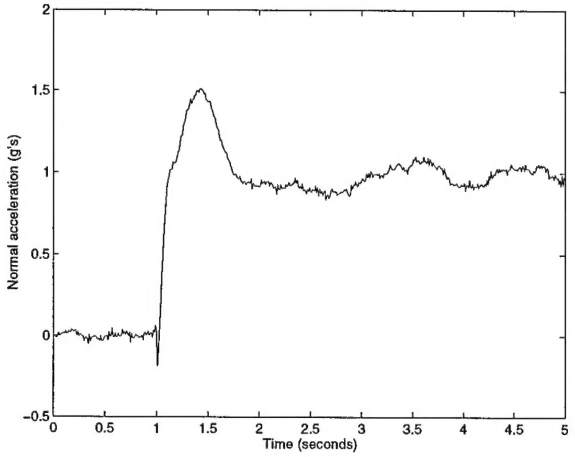

Figure 13: Step response with noise, design point 15

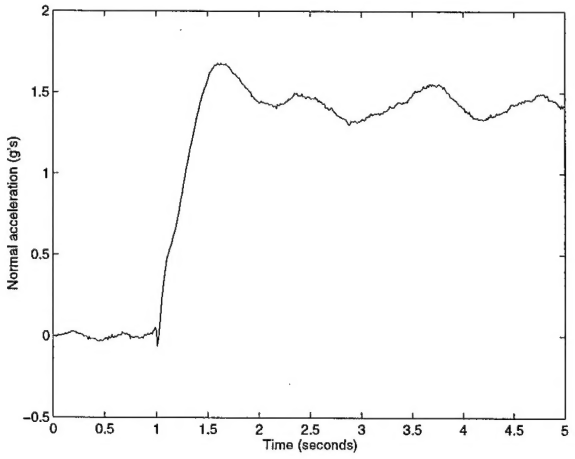

Figure 14: Step response with noise, design point 24

approach to fixed-order (sub-optimal) H_2/L_1 optimization for continuous systems. Two methods were developed to numerically approximate the L_1 norm of a continuous system and analytical gradients of the L_1 problem with respect to the state-space elements of the controller were derived. The H_2/L_1 algorithm was applied to an AFTI F-16 longitudinal control problem which clearly demonstrated the benefits of mixing H_2 and L_1 optimization methods. Additional research is needed to further refine the calculation the L_1 norm.

9 Appendix

This section contains the state-space descriptions of the aircraft model shown in Figure 2.

The four states in the longitudinal model, G_p are forward speed (u in ft/sec), angle of attack (α in *radians*, pitch angle (θ in *radians*), and pitch rate (q in *radians/sec*). The input to G_p is the stabilator deflection (δ_e in *radians*) and the output is the normal acceleration (n_z in $g's$). G_p is given by

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -1.485e-2 & 3.738e+1 & \cdots \\ -8.000e-5 & -1.491e+0 & \cdots \\ 0.000e+0 & 0.000e+0 & \cdots \\ -3.600e-4 & 9.753e+0 & \cdots \end{bmatrix}$$

$$\begin{matrix} -3.220e+1 & -1.794e+1 \\ -1.300e-3 & 9.960e-1 \\ 0.000e+0 & 1.000e+0 \\ 2.900e-4 & -9.600e-1 \end{matrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} +$$

$$\begin{bmatrix} 2.140e-3 \\ -1.880e-1 \\ 0.000e+0 \\ -1.904e+1 \end{bmatrix} \delta_e$$

$$n_z = \begin{bmatrix} 1.500e-3 & 3.5264e+1 & \cdots \end{bmatrix}$$

$$\begin{matrix} 2.720e-2 & -3.340e-1 \end{matrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} +$$

$$\begin{bmatrix} -4.366e+0 \end{bmatrix} \delta_e$$

The input to G_a is the commanded stabilator deflection (δ_{e_c} in *radians*) and the output is the stabilator deflection. G_a is given by

$$\dot{x}_a = \begin{bmatrix} -2.000e+1 \end{bmatrix} x_a + \begin{bmatrix} 2.000e+1 \end{bmatrix} \delta_{ec}$$

$$\delta_e = \begin{bmatrix} 1.000e+0 \end{bmatrix} x_a + \begin{bmatrix} 0.000e+0 \end{bmatrix} \delta_{ec}$$

The input to G_t is normal acceleration and the output is the delayed normal acceleration (n_{z_d} in $g's$). G_t is given by

$$\dot{x}_t = \begin{bmatrix} -4.000e+1 \end{bmatrix} x_t + \begin{bmatrix} 1.000e+0 \end{bmatrix} n_{z_d}$$

$$n_{z_d} = \begin{bmatrix} 8.000e+1 \end{bmatrix} x_t + \begin{bmatrix} -1.000e+0 \end{bmatrix} n_z$$

The input to G_w (used in simulation only) is a unit strength white Gaussian noise (w_1) and the output

is the wind noise (ζ_1). G_w is given by

$$\begin{aligned} \dot{x}_w &= \left[\ -6.700e+0\ \right] x_w + \left[\ 1.870e-3\ \right] w_1 \\ \zeta_1 &= \left[\ 1.000e+0\ \right] x_w \end{aligned}$$

The input to G_m (used in simulation only) is a unit strength white Gaussian noise (w_2) and the output is the measurement noise (ζ_2). G_m is given by

$$\begin{aligned} \dot{x}_m &= \left[\ -1.000e+1\ \right] x_m + \left[\ 4.000e-4\ \right] w_2 \\ \zeta_2 &= \left[\ 1.000e+0\ \right] x_m + \left[\ 4.000e-4\ \right] w_2 \end{aligned}$$

References

[1] D. Mustafa and K. Glover, "Controllers which Satisfy a Closed-Loop H_∞-norm Bound and Maximize an Entropy Integral," in *Proceedings of the 27th Conference on Decision and Control*, (Austin TX), pp. 959–964, December 1988.

[2] D. S. Bernstein and W. H. Haddad, "LQG Control with an H_∞ Performance Bound: A Riccati Equation Approach," *IEEE Trans. Auto. Control*, vol. AC-34, pp. 293–305, March 1989.

[3] J. Doyle, K. Zhou, and B. Bodenheimer, "Optimal Control with Mixed H_2 and H_∞ Performance Objectives," in *Proceedings of American Control Conference*, (Pittsburgh PA), pp. 2065–2070, June 1989.

[4] D. Mustafa and K. Glover, "Minimum Entropy H_∞ Control," in *Lecture Notes in Control & Information Sciences*, Springer-Verlag, 1990.

[5] H. H. Yeh, S. S. Banda, and B.-C. Chang, "Necessary and Sufficient Conditions for Mixed H_2 and H_∞ Optimal Control," in *Proceedings of the 29th Conference on Decision and Control*, (Honolulu HI), pp. 1013–1017, December 1990.

[6] K.Zhou, J. Doyle, K. Glover, and B. Bodenheimer, "Mixed H_2 and H_∞ Control," in *Proceedings of the American Control Conference*, (San Diego CA), pp. 2502–2507, May 1990.

[7] P. P. Khargonekar and M. A. Rotea, "Mixed H_2/H_∞ Control: A Convex Optimization Approach," *IEEE Trans. Auto. Control*, vol. AC-36, pp. 824–837, July 1991.

[8] D. E. Walker, *H_2 Optimal Control with H_∞, μ, and L_1 Constraints.* PhD thesis, Air Force Institute of Technology, WPAFB, OH, June 1994.

[9] A. Megretski, "On the Order of Optimal Controllers in the Mixed H_2/H_∞ Control," in *Proceedings of the 33rd Conference on Decision and Control*, (Lake Buena Vista FL), pp. 3173–3174, December 1994.

[10] D. E. Walker and D. B. Ridgely, "A New Numerical Method for the General Mixed H_2/H_∞ Optimal Control Problem." *Preprint*, 1994.

[11] D. E. Walker and D. B. Ridgely, "Fixed Order Mixed H_2/H_∞ Optimization with Multiple H_∞ Constraints." Submitted to the *AIAA Journal of Guidance, Control and Dyanmics*, 1994.

[12] U. L. Ly and E. Schomig, "A New Approach to Mixed H_2/H_∞ Controller Synthesis Using Gradient Based Parameter Optimization Methods." University of Washington, Final Technical Report under NASA grant contract NAG-2-629, 1993.

[13] M. Vidyasagar, "Optimal Rejection of Persistent Bounded Disturbances," *IEEE Trans. Auto. Control*, vol. AC-31, pp. 527–534, Jun 1986.

[14] M. A. Dahleh and J. B. Pearson, "ℓ_1 Optimal Feedback Controllers for MIMO Discrete-Time Systems," *IEEE Trans. Auto. Control*, Apr 1987.

[15] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach.* Prentice-Hall, 1994.

[16] P. Voulgaris, "Optimal H_2/ℓ_1 Control: The SISO Case," in *Proceedings of the 33rd Conference on Decision and Control*, (Lake Buena Vista FL), pp. 3181–3186, December 1994.

[17] D. R. Jacques and D. B. Ridgely, "A Fixed-Order, Mixed-Norm Control Synthesis Method for Discrete Linear Systems," in *Proceedings of the American Control Conference*, (Seattle, WA), 1995.

[18] Nelson, "Simplified Calculation of Eigenvector Derivatives," *AIAA Journal*, vol. 14, pp. 1201–1205, September 1976.

[19] D. R. Jacques, D. B. Ridgely, R. A. Canfield, and M. S. Spillman, "A MATLAB Toolbox for Fixed-Order, Mixed-Norm Control Synthesis," in *Proceedings of the 4th IEEE Conference on Control Applications*, (Albany, NY), 1995.

[20] K. Schittkowski, "NLPQP: A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems," in *Annals of Operation Research*, pp. 485–500, J. C. Baltzer A. G., Scientific Publishing Company, 1985/6.

[21] J. Luke, "Flight Controller Design Using Mixed H_2/H_∞ Optimization with a Singular H_∞ Constraint," Master's thesis, Air Force Institute of Technology, WPAFB, OH, December 1993.

[22] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dyanmic Systems.* Addison-Wesley Publishing Company. Inc., 1991.

[23] D. B. Ridgely and S. S. Banda, *Introduction to Robust Multivariable Control.* AFWAL-TR-85-3102, USAF, 1986.